

TREE STRUCTURAL DRAWING GENERATING SYSTEM

Patent Number: JP2148173
Publication date: 1990-06-07
Inventor(s): DEGUCHI NAOMI
Applicant(s): FUJI XEROX CO LTD
Requested Patent: ☐ JP2148173
Application Number: JP19880300648 19881130
Priority Number(s):
IPC Classification: G06F15/40; G06F15/62
EC Classification:
Equivalents: JP2789618B2

Abstract

PURPOSE:To constitute the above system so that even if a tree structure is complicated, it can be handled by a satisfactory response by setting a partial tree structure and storing its data by separating it from other tree structure part, and storing the data to a memory area by dividing it.

CONSTITUTION:The system is provided with a node input means 11, a partial tree structure setting means 12, a tree structure data store means 13, a mark giving means 14, a tree structure generating means 15, and a tree structure display means 16. In this state, when the tree structure is complicated, a partial tree structure in which a prescribed node is the apex is set as one node (slave node), the display is simplified thereby, and also, by storing detailed data related to the partial tree structure in an area being different from data of other structure part than the structure, the data is dispersed. In such a way, even if the tree structure is complicated as a whole, it can be processed by a satisfactory response.

Data supplied from the esp@cenet database - I2

⑫ 公開特許公報(A)

平2-148173

⑤Int.Cl.⁵G 06 F 15/40
15/62

識別記号

5 0 0 C
3 2 0 K

庁内整理番号

7313-5B
8125-5B

⑬公開 平成2年(1990)6月7日

審査請求 未請求 請求項の数 1 (全1頁)

⑭発明の名称 木構造図作成システム

⑯特 願 昭63-300648

⑰出 願 昭63(1988)11月30日

⑱発明者 出口 直 実 東京都渋谷区代々木3丁目57番6号 グランフォーレ 富士ゼロックス株式会社内
 ⑲出願人 富士ゼロックス株式会社 東京都港区赤坂3丁目3番5号
 ⑳代理人 弁理士 山内 梅雄

明 細 書

1. 発明の名称

木構造図作成システム

2. 特許請求の範囲

木構造を作成するための節としてのそれぞれのノードを、ノード間の連鎖関係を示しながら入力するノード入力手段と、

作成される木構造の一部について、所定のノードを頂点として木構造状に切り取った形の部分木構造を設定する部分木構造設定手段と、

部分木構造設定手段によって設定された部分木構造に関するデータをそれぞれ他の構造部分のデータとは独立した領域に格納する木構造データ格納手段と、

部分木構造ごとにそれらを表わす識別マークを付与するマーク付与手段と、

前記ノード入力手段から入力されたノードのうち前記部分木構造に属さないものについてはノード単位の連鎖関係で木構造を組み立て、前記部分木構造に属するものについては部分木構造ごとに

1つのノードを割り当てて前記木構造に連結する木構造作成手段と、

この木構造作成手段によって作成された木構造のうち部分木構造を表わすノードとしての子ノードについてはマーク付与手段で付与されたマークをノード表示用に表示する木構造表示手段とを具備することを特徴とする木構造図作成システム。

3. 発明の詳細な説明

「産業上の利用分野」

本発明は、階層構造をもったデータを基にして木構造図を作成し、これを表示するための木構造図作成システムに関する。

「従来の技術」

階層構造をもったシステムの計画あるいは設計の際には、従来から木構造が一般に用いられている。これは、木構造がモジュール構成あるいはデータ構造の記述に適しているからであり、ソフトウェアの設計段階でも木構造は広く使用されている。また機械設計の分野でも、総組立図、部分組

立図、部品図のように、これらが階層構造をもっている場合には、図面や部品の構成をより明確に表現するために木構造の活用が注目されている。

ところで、木構造(tree structure)における“木”とは、1つ以上の節(ノード)の有限集合をいう。木構造は例えば第15図に示す各ノードA、B、……Gを枝(エッジ)で連結したものであり、階層的な関係を表現するのに適した構造である。この例では、ノードAが最上層(第1レベル)に位置し、ノードGが最下層(第4レベル)に位置する。

このような木構造は、従来では手書きによって、あるいはテンプレートを使用したなぞり書きによって作成していた。第16図は、このうちテンプレートを使用したモジュール構成の一例を表わしたものである。この例の場合、メインフレームMは、3つのサブフレームS1～S3によって構成されており、このうちの第2のフレームS2は更に3つのサブフレームS21～S23によって構成されている。

その作業が大変であった。また、一度その木構造を変更してしまうと前の状態が消し去られることになり、設計変更等の履歴を残すことが困難であった。

これに対して、コンピュータを使用した木構造図作成システムあるいは編集システムでは、図面の構成リスト等のリストを出力することができる。従って第17図に示したように、このようなテキスト(電文)イメージに既に説明したレベル番号を付加するか、その表示位置のカラムを桁下がりにして階層表示を行うこともできる。しかしながら、これらは階層的な構造を明瞭に表示するものではなく、このような明瞭な表示を望む場合には、第18図に示したようにテンプレートを使用した作業に頼らざるを得なかった。

一方、コンピュータの表示装置や出力装置におけるグラフィック化は特に目覚ましいものがあり、近年ではグラフィックイメージの表示や出力を可能とする木構造図作成システムあるいは編集システムが出現している。

一方、近年では、コンピュータを使用した木構造図作成システムあるいは編集システムの開発も行われている。

第17図はこのような従来の木構造図作成システムによって作成された木構造を表わすためにその付属のプリンタから出力された部品の構成リストの例を示しており、第18図はこの構成リストを基にしてテンプレートを使用して作成した木構造を示している。このうち第17図では各部品の階層が“LEVEL”番号として“1”から“4”まで表示されており、第18図ではこれを基にして階層表現が行われている。部品によっては、例えば部品番号“1P100001”の“ボルト1”のように異なったノードに共通して接続されるものも存在する。

「発明が解決しようとする課題」

さて、第16図および第18図に示したようなテンプレートを使用した木構造の表示によると、設計変更が行われるたび等に、消しゴムや修正液等の修正手段を用いて図を書き直す必要があり、

第19図は、このようなグラフィック機能を持ったシステムによる表示の一例を表わしたものである。また、他の木構造についてその各ノードを丸付数字①、②等で抽象化して表わすと第20図に示すようなものとなる。ここで階層の最上位はルート(root)と呼ばれている。

ところでこの第20図に示した例ではノードの数が6つである。このようにノードの数が比較的小さい場合には、これを第19図に示したようなグラフィックで表現する場合であってもその処理に特別の時間を要しない。ところが、大規模なシステムや複雑な機械の部品を管理する場合等では、1000ノードあるいはこれ以上のノード数の木構造が要求されることになる。このような規模の木構造を扱う場合には、処理側のCPU(中央処理装置)に比較的大容量のメモリを要求することになる。また、CPU側の処理時間もノードの数に応じて増加する。

第21図はこの処理時間の増加の様子の一例を表わしたものである。図で横軸は木構造を構成す

るノードの数であり、縦軸は処理時間(秒)である。図に示すように、ノードの数が増加すると処理時間は指数関数的に増加することがわかる。このように木構造が複雑になるとその処理時間が増加する。従って、一般にノードが1000以上の木構造になってくると、ワークステーションのCPUの能力では、会話形式に支障を生じさせない程度のレスポンスで木構造を取り扱うことができなくなるという問題があった。

これを更に具体的に説明する。

第22図は、木階層構造における内部表現の一例を表わしたものである。ここではノードIDとして4つのノード“01”から“04”までを示している。「親ノード数」とは、自己のノードに対する1つ前(上位)のノードの数をいい、最上位のノードであればこれが“0”となる。「子ノード数」とは、自己のノードに対する1つ後(下位)のノードの数をいう。1つのノードに対して親ノードがN個あれば親ノード用のアドレスがN個必要であり、子ノードがM個あれば子ノード用

のアドレスがM個必要となる。この例では、説明を簡単にするために1つのノードに対する親ノードおよび子ノードの最大数をそれぞれ“3”としている。

ところが、現実には親ノードおよび子ノードの数をこのように限定することができない場合が多い。試しに、1000件の処理が可能なメモリサイズと、2000件の処理が可能なメモリサイズを計算してみることにする。第22図に示したノードIDもノード数も共に2000件までなので、2バイトで表現可能であり、ノード名を24桁までとして24バイト、アドレスを4バイトに設定すると、次の第1表のようになる。

第1表

ノードID	2バイト
ノード名	24バイト
親ノード数	2バイト
親ノードID	4バイト×リンク可能数
子ノード数	2バイト
子ノードID	4バイト×リンク可能数

ここで、リンク可能数が問題となる。第23図は、1000件の場合における親ノード数に関して一番極端なリンク構造を表わしたものである。このように、極端な例では1つのノード“1”の下に998個のノード“2”～“999”が存在し、これらの下に1つのノード“1000”が存在する。よって親ノードの最大数はノード“1000”から見た親ノード数であり、998個となる。

これに対して第24図では、1000件の場合における子ノード数に関して一番極端なリンク構造を表わしたものである。この例では、1つのノード“1”の下に999個のノード“2”～“1000”が存在する。このように、子ノードの最大数は1000個となる。

従って、1000件の処理が可能なメモリサイズは、1ノードのレコードサイズについて次の第2表のようになる。

(以下余白)

第2表

ノードID	2バイト
ノード名	24バイト
親ノード数	2バイト
親ノードID	4バイト×998
子ノード数	2バイト
子ノードID	4バイト×999

すなわち、この場合には合計で8018バイトとなり、これが1000ノードについて用意されなければならないので、全体として合計8018メガバイトの記憶容量が必要になることになる。

2000件の処理が可能なメモリサイズについては、同様にして親ノードの最大数が“1998”、子ノードの最大数が“1999”となり、メモリサイズは、1ノードのレコードサイズについて次の第3表のようになる。

(以下余白)

第3表

ノードID	2 バイト
ノード名	24 バイト
親ノード数	2 バイト
親ノードID	4 バイト × 1998
子ノード数	2 バイト
子ノードID	4 バイト × 1999

すなわち、この場合には合計で16018バイトとなり、これが2000ノードについて用意されなければならないので、全体として合計32.036メガバイトの記憶容量が必要になることになる。これは1000件の処理の場合に比べて約4倍のメモリサイズを意味する。このように件数が増えれば、それに伴ってメモリサイズも大幅に増加していく。これ故、木構造が複雑化すればするほど、レスポンスが悪くなることになる。

そこで本発明の目的は、木構造が複雑化しても良好なレスポンスでこれを取り扱うことのできる木構造図作成システムを提供することにある。

与手段で付与されたマークをノード表示用に表示する木構造表示手段16とを木構造図作成システムに具備させる。

すなわち本発明では、木構造が複雑化してきたような場合には、所定のノードを頂点とする部分木構造を1つのノード(子ノード)とし、これにより表示の簡素化を図ると共に、部分木構造に関する詳細なデータをこれ以外の構造部分のデータとは別の領域に格納することにして、データの分散を図り、全体として木構造が複雑化しても良好なレスポンスでこれを処理できるようにする。

「実施例」

以下実施例につき本発明を詳細に説明する。

第2図は本発明の一実施例としての木構造図作成編集システムについて、その構成の概要を表わしたものである。木構造図作成編集システムは、CPU(中央処理装置)21を備えている。このCPU21はデータバス等のバス22によって次の各部と接続されている。

(i) RAM 23:

「課題を解決するための手段」

本発明では、第1図に原理的に示すように、木構造を作成するための節としてのそれぞれのノードを、ノード間の連鎖関係を示しながら入力するノード入力手段11と、作成される木構造の一部について、所定のノードを頂点として木構造状に切り取った形の部分木構造を設定する部分木構造設定手段12と、部分木構造設定手段12によって設定された部分木構造に関するデータをそれぞれの構造部分のデータとは独立した領域に格納する木構造データ格納手段13と、部分木構造ごとにそれらを表わす識別マークを付与するマーク付与手段14と、ノード入力手段11から入力されたノードのうち部分木構造に属さないものについてはノード単位の連鎖関係で木構造を組み立て、部分木構造に属するものについては部分木構造ごとに1つのノードを割り当てて木構造に連結する木構造作成手段15と、この木構造作成手段15によって作成された木構造のうち部分木構造を表わすノードとしての子ノードについてはマーク付

このシステムを動作させるための各種プログラムを格納したり、データを一時的に格納するためのランダム・アクセス・メモリである。

(ii) ハードディスク装置24:

RAM 23に読み出すためのプログラムを格納したり、作成された木構造に関するデータを格納するための記憶装置である。装置によってはフロッピーディスク等の記憶媒体にプログラムを格納したり、作成された木構造に関するデータをこれらの記憶媒体に格納するようにしてもよい。

(iii) キーボード25:

システムの操作上必要な各種データを入力する入力装置である。

(iv) マウス26:

ポインティング・デバイスの1種であり、このシステムではキーボード25を介してバス22に接続されている。木構造を形成する個々の項目の位置等を入力する際に使用される。

(v) グラフィックディスプレイ27:

グラフィック化されたデータを表示することの

できるディスプレイである。木構造の作成や編集を行う際に用いられる。

(vi) グラフィックプリンタ: 28

木構造の作成過程や編集結果を印刷することのできるプリンタである。

ところで、この木構造図作成編集システムではカードという概念を使用している。この明細書でカードとは、各ノードの要素をいう。カードには、それぞれの部品やシステム等の情報を有するノートカードと、木構造自体を表示するブラウザカードの2種類がある。ノートカードは、例えば部品の情報を有するものであれば部品カードという愛称で呼ぶこともできるし、図面の情報を有するものであれば図面カード、またアイディアに関する情報を有するものであればアイディアカードというような愛称を付けて呼ぶこともできる。

第3図は、このうちの部品カードの一例を表わしている。この部品カードはグラフィックディスプレイ27上に開かれたウィンドウ31に割り当てられ、カードの名称32と内容33が表示され

る。ウィンドウ31の右上端部には“閉じる”という文字が表示されている。マウス26を操作することでこの部分を図示しないカーソルで指示すると、ウィンドウ31が閉じるようになっている。ウィンドウ31の右端部には、その上下に三角形のマーク35、36が配置されており、これらのいずれかの部分をマウス26によって指定することでウィンドウ31内の表示内容がスクロールすることになる。この例では、ウィンドウ31内に、部品名、材質、単価、重量および業者に関するデータが表示されている。

第4図は他の例としてアイディアカードのウィンドウを表わしたものである。この例に示したアイディアは、その内容をウィンドウ37内に一度に表示することができない。そこで、スクロール用の三角形のマーク35、36をマウス26で適宜選択して必要な箇所のデータを読み取ることになる。

第5図～第7図は、ブラウザカードのウィンドウについてその表示例を表わしたものである。ブ

ブラウザカードは木構造を表示するためのものであり、木構造が比較的単純であれば第5図に示したように全体の構造をウィンドウ38内に表示することができる。

第6図および第7図は、第5図に示した木構造を2つに分割して表現するためのブラウザカードを表わしたものである。このうち第6図に示したブラウザカードは便宜上“Aブラウザカード”と呼ぶことにする。Aブラウザカードの木構造データは、Aデータファイルと便宜的に呼ぶ1つのデータファイル内に格納される。第6図に示した木構造では、“部分組図Ⅲ”のノードを頂点とした部分木構造についての情報を“Bブラウザカード”としてまとめている。この第6図でBブラウザカード以外のノードについては、図面カードと呼ばれるカード(図示せず)にその内容が格納されることになる。

第7図がBブラウザカードの木構造データを表わしたものである。Bブラウザカードの木構造データ(部分木構造データ)は、Bデータファイル

と便宜的に呼ぶ1つのデータファイル内に格納される。すなわち、この例では1つの木構造が2つに分割されて表示がし易くなったばかりでなく、同様に2つのデータファイルにデータを分割して格納することにして、それぞれのファイルにおけるデータの総量を減少させ、その処理速度の向上を図ることができる。

第7図に示した例では、Bブラウザカードの中に更に他のブラウザカードとして“Cブラウザカード”が存在している。このように分割されたブラウザカード内に更にブラウザカードが存在してもよく、これにより、複数のデータファイル自身が木構造の階層を持つことが可能になる。

この木構造図作成編集システムで、以上説明した木構造の表示を行う際の操作を説明すると次のようになる。

まずオペレータは、グラフィックディスプレイ27(第2図)上に表示された各種アイコンの中からAブラウザカードに対応するアイコンを選択する。すると、第6図に示したようにウィンドウ

38Aがオープンし、Aブラウザカードがこれに表示される。オペレータがこの木構造で1つのノードとして表示されたBブラウザカードについてその内容を知りたいときには、マウス26(第2図)を操作して図示しないカーソルをBブラウザカードのノードに合わせる。そしてこのマウス26の図示しないボタンをクリックすると、第7図に示したように新たなウィンドウ38Bがオープンし、ここにBブラウザカードの内容が表示されることになる。

このように、本実施例のシステムでは、木構造がいかに巨大であろうとも、システムが一度に処理するノード数は表示あるいは参照したい部分のブラウザカードのノード数で済むので、良好なレスポンスを得ることができる。もちろん、それぞれのブラウザカードの部分木構造データは、各々別個の記憶領域に分散的に格納されることになる。従って、CPUが一度に処理するデータの量はそれほど膨大とはならず、CPUの必要とするメモリの容量もそれほど大きなものとはならない。更

に、木構造の表示にブラウザカードを用いることで部分木構造をそれぞれ1つずつのノードとして表現することができるので、木構造すべてをディスプレイに表示する場合と異なり、特に大きなウィンドウを必要とすることもない。また、巨大な木構造図に関するデータであっても、これらを格納するデータファイルはブラウザカード単位等で分散管理することもできるし、フロッピーディスク、磁気テープ等の非常駐マウントのメディアに分散させることも可能である。このようなことから、このシステムでは木構造データの格納について物理的な容量の制約が存在せず、論理的にほぼ無限の空間を利用することができる。

「変形例」

第8図は、他のブラウザカードの例を表わしたものである。

このブラウザカード“1M100001”には、木構造のすべてが表示されている。第9図および第10図は、これを本発明の木構造図作成システムで2つのブラウザカードに分けて表示した例を

表わしたものである。この例では、部分木構造を表示したブラウザカード“1P100002”を第9図で二重枠で表示している。このような表示を行う代わりに、例えば第11図に示すようにノード名の後に右方向の矢印(→)を表示するようにしてもよい。

第12図は、この変形例における木構造の表示のための制御を表わしたものである。この表示処理に際してこの変形例では、第2図に示した木構造図作成システムと同等のシステムを使用するが、以下の説明では便宜的に第2図に使用した番号を使用することにする。

さて、このシステムが表示処理を行うモードでは、CPU21は、処理すべきデータファイルの指示が入力されるのを待機している(ステップ①)。オペレータがキーボード25からデータファイルの指示を入力すると(Y)、CPU21はハードディスク24上に配置されたブラウザカードデータファイル内の処理データファイルを読み出す(ステップ②)。そして、木構造における最

上位ノードを見つけることになる(ステップ③)。最上位ノードが見つかったら、これを処理ノードに設定する(ステップ④)。そして、同じくハードディスク24上に配置された属性データファイルを読み出し(ステップ⑤)、前記した処理ノードをグラフィックディスプレイ27上に表示する(ステップ⑥)。

次にCPU21は、処理ノードに対する子ノードが存在するかどうかの判別を行う(ステップ⑦)。子ノードが無ければ(N)、この時点で処理が終了する(エンド)。子ノードが存在すれば(Y)、子ノードをグラフィックディスプレイ27上に表示するための処理を行う(ステップ⑧)。そして、子ノードが存在するかどうかの判別作業が行われる(ステップ⑦)。以下同様にして子ノードが存在する限り同様の作業が繰り返されることになる(ステップ⑦、⑧)。

第13図は、第12図のステップ⑧で示した処理の具体的な内容を表わしたものである。子ノードの表示処理を行うに際しては、まずそれがブラ

ウザカードに格納されているかどうかの判別が行われる(第13図ステップ①)。ブラウザカードに格納されていなければ(N)、属性データファイルの読み出しが行われる(ステップ②)。そして、処理ノードをグラフィックディスプレイ27上に表示する(ステップ③)。次にCPU21は、処理ノードに対する子ノードが存在するかどうかの判別を行う(ステップ④)。子ノードがあれば(Y)、この第13図に示した子ノードの表示処理と同様の処理が行われる(ステップ⑤)。そして、この処理されたノードに対する子ノードが存在するかどうかの判別作業が行われる(ステップ④)。以下同様にして子ノードが存在する限り同様の作業が繰り返されることになる(ステップ④、⑤)。

ステップ④で子ノードが存在しなければ(N)、次のノードが存在するかどうかの判別が行われる(ステップ④)。存在すれば(Y)、ステップ①に戻って前記したと同様の作業が繰り返されることになる。次のノードが存在しなかった場合には、

し、第9図で示した“1P100002”のように表記されたブラウザカードが指示された場合には(Y)、第10図に表わしたように子ブラウザカードの探索処理が行われる(ステップ⑦)。ブラウザカード以外のノードが指示された場合には(ステップ⑥; N)、そのノードについての属性データファイルが読み出される(ステップ⑧)。

ここで子ブラウザカードの探索処理(ステップ⑦)は、この第14図に示した流れと全く同一となる。すなわち、子ブラウザカードの探索処理ではまず子に相当するブラウザカードのデータファイルが読み出され、その最上位ノードが見つかり、木走査が行われ、その結果がグラフィックディスプレイ27(第2図)上に表示される。この状態でオペレータがどのノードが選択されるかの監視が行われ、ブラウザカードが選択されれば、ステップ⑦で示したと同様の処理が行われることになる。これ以外のノードが選択されれば、その属性データが読み出されることになる。

以上のようにして、木構造の探索が行われる。

親のノードに戻り、子のノードとして他のノードが存在するかどうかの判別が行われることになる(第12図ステップ⑦)。

第14図は、木構造の探索を行う場合のその制御の様子を表わしたものである。

木構造の所定のノードを探索する状態でCPU21は、処理すべきデータファイルの指示が入力されるのを待機している(ステップ①)。オペレータがキーボード25からデータファイルの指示を入力すると(Y)、CPU21はハードディスク24上に配置されたブラウザカードデータファイル内の処理データファイルを読み出す(ステップ②)。そして、木構造における最上位ノードを見つけることになる(ステップ③)。最上位ノードが見つかったら、木走査を行う(ステップ④)。これにより木構造が復元されるので、これを基にして第9図に示したようなブラウザカードウィンドウを表示する(ステップ⑤)。この状態で、CPU21は、オペレータがどのノードを探索のために指示するかを待機する(ステップ⑥)。も

「発明の効果」

このように本発明によれば、部分木構造を設定しそのデータを他の木構造部分と切り離して記憶することにしたので、メモリ領域に対するデータの格納を実質的に分割して行うことができ、データの処理速度の向上を図ることができる。また、木構造表示手段が一度に表示するデータの数を減少させることができるので、所望のノードを迅速に探し出すことができ、この意味でも探索のスピードが向上する。

4. 図面の簡単な説明

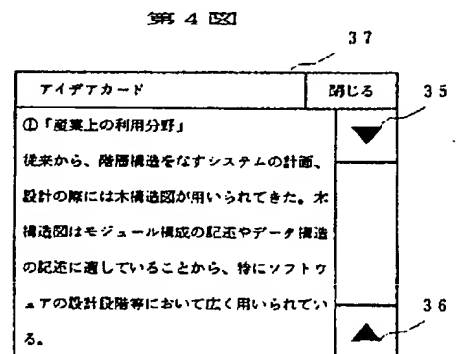
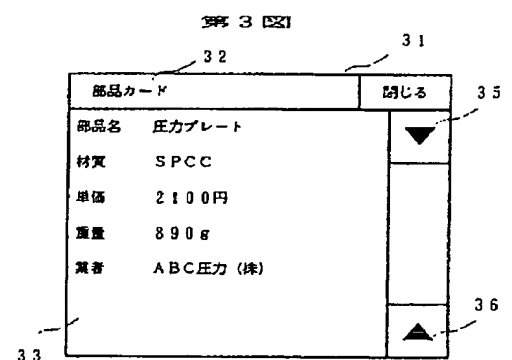
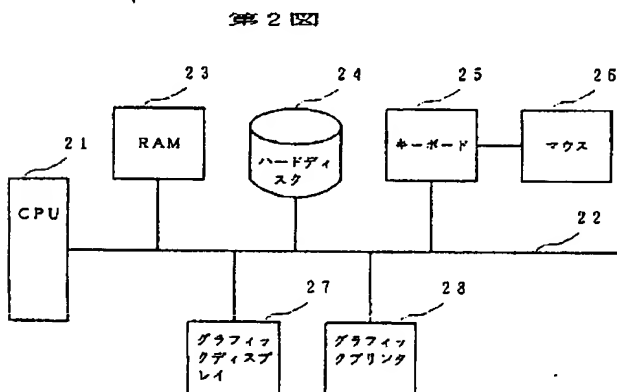
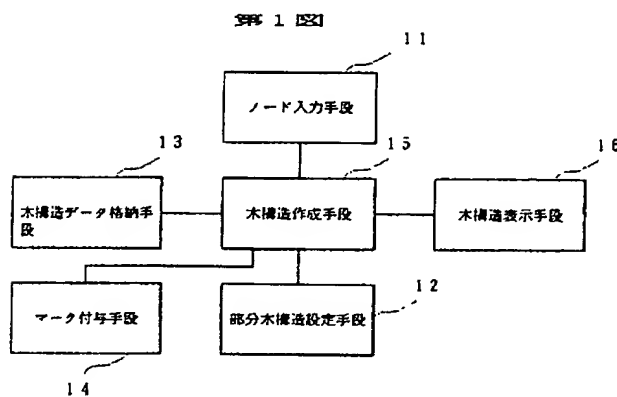
第1図は本発明の原理を示す原理図、第2図～第7図は本発明の一実施例を説明するためのもので、このうち第2図は木構造図作成編集システムについてその構成の概要を表わしたブロック図、第3図はグラフィックディスプレイ上に表示された部品カードの一例を示す平面図、第4図はグラフィックディスプレイ上に表示されたアイディアカードの一例を示す平面図、第5図～第7図はブラウザカードのウィンドウの表示例を示す平面図、

第8図～第14図は本発明の変形例を説明するためのもので、このうち第8図はブラウザカードの一例を示す平面図、第9図は子の木構造を一部用いて第8図に示した木構造を表わしたブラウザカードの平面図、第10図は子の木構造のブラウザカードを示す平面図、第11図は部分木構造であることを示すノードの他の表示例を示す平面図、第12図はこの変形例における木構造表示のための制御の様子を示す流れ図、第13図は第12図のステップ⑧で示した処理の具体的な内容を表わした流れ図、第14図は木構造の探索を行う場合の制御を表わした流れ図、第15図は木構造の一例を示す説明図、第16図はテンプレートを使用して表現した木構造の一例を示す説明図、第17図は部品の構成リストの一例を示す平面図、第18図はこの構成リストを基にしてテンプレートを使って作成した木構造の平面図、第19図はグラフィック機能を持った従来のシステムで作成した木構造の表示例を示す平面図、第20図はノードの数が6つの場合の木構造の一例を示す説明図、

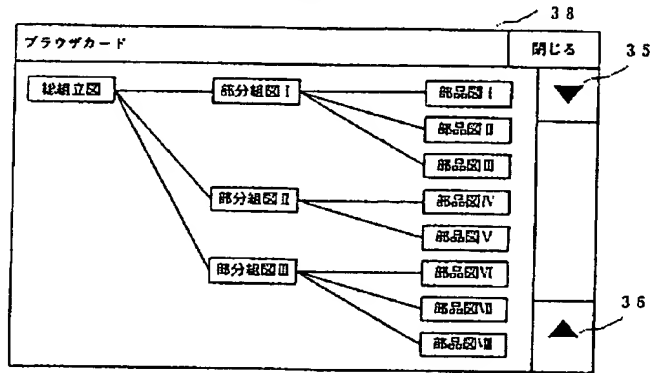
第21図はノードの数とCPUの処理時間との関係を示す特性図、第22図は木階層構造における内部表現の一例を表わした説明図、第23図は1000件の処理が可能なメモリサイズにおける親ノードが最大になる状態を示したリンク構造図、第24図は同様のメモリサイズにおける子ノードが最大になる状態を示したリンク構造図である。

- 11 …… ノード入力手段、
 12 …… 部分木構造設定手段、
 13 …… 木構造データ格納手段、
 14 …… マーク付与手段、
 15 …… 木構造作成手段、
 16 …… 木構造表示手段、
 21 …… CPU、
 23 …… RAM、
 24 …… ハードディスク装置、
 25 …… キーボード、
 26 …… マウス、
 27 …… グラフィックディスプレイ、
 28 …… グラフィックプリンタ、
 31～38 …… ウィンドウ。

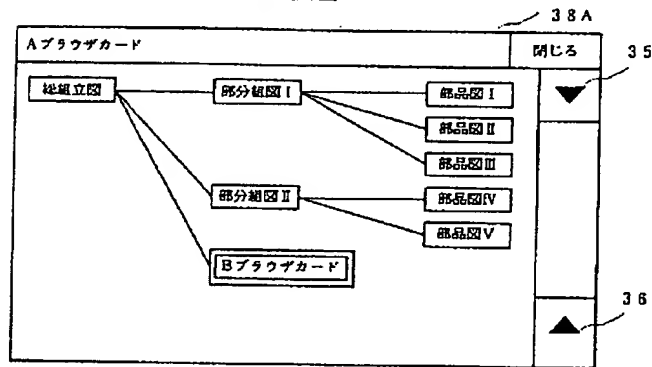
出 願 人 富士ゼロックス株式会社
 代 理 人 弁 理 士 山 内 梅 雄



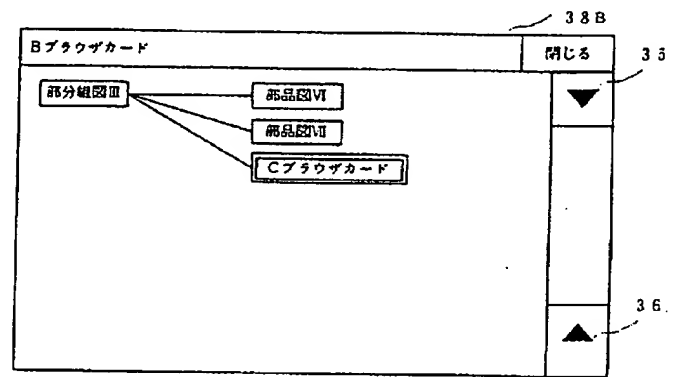
第 5 図



第 6 図



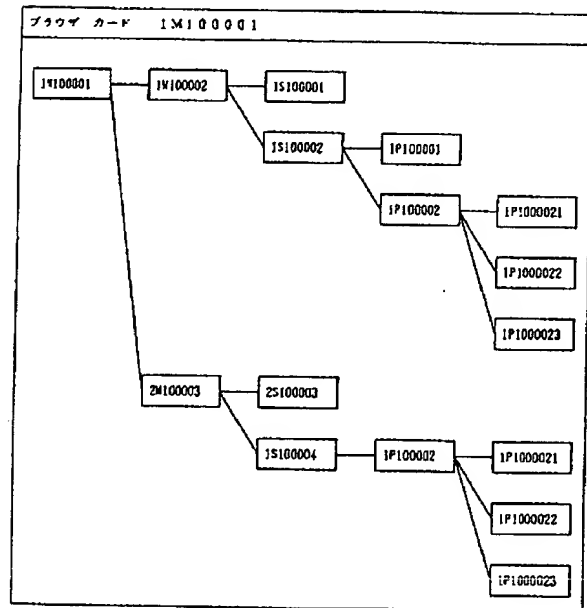
第 7 図



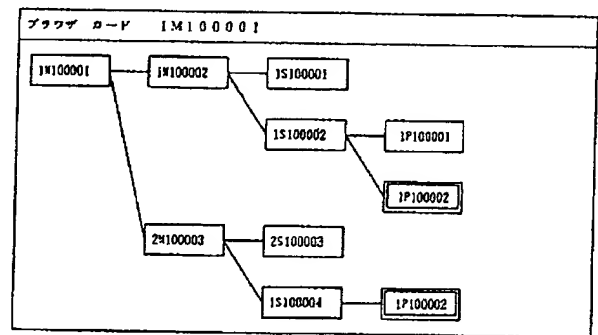
第 11 図

IP100002-

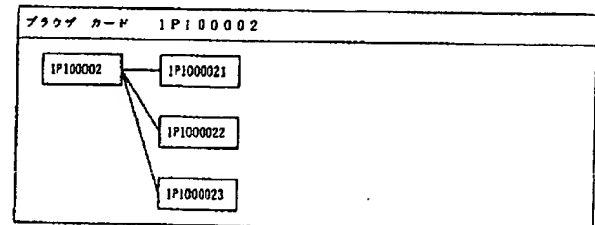
第 8 図



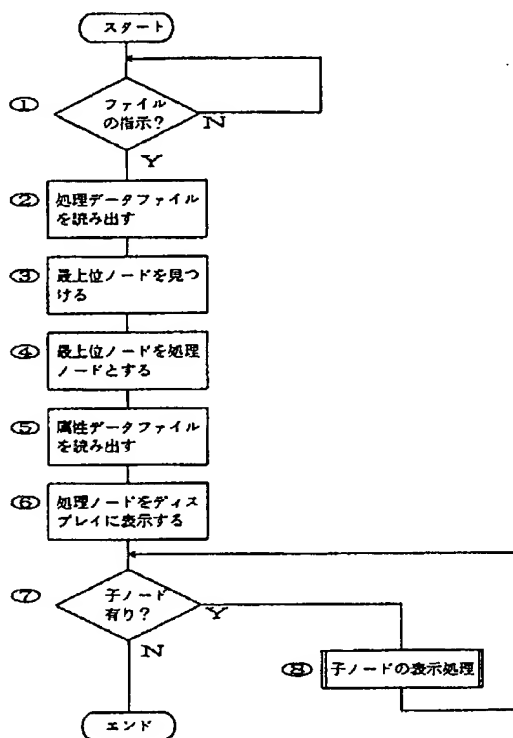
第 9 図



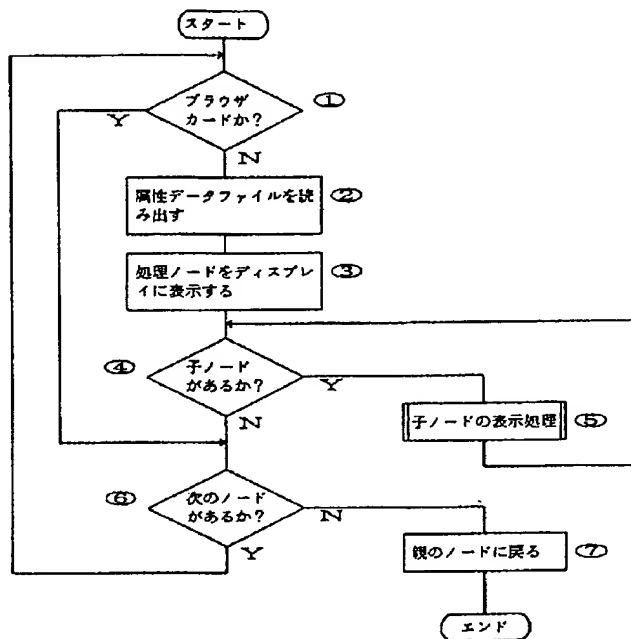
第 10 図



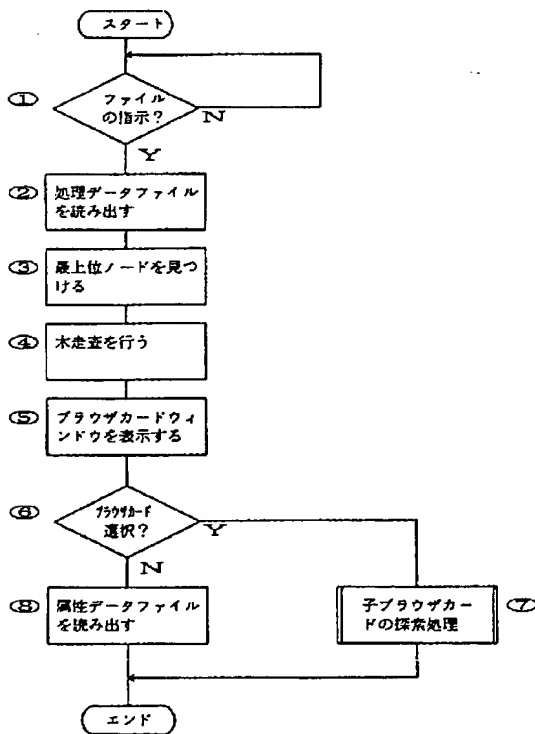
第 1 2 図



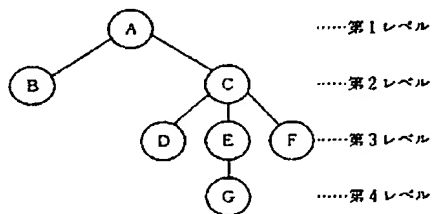
第 1 3 図



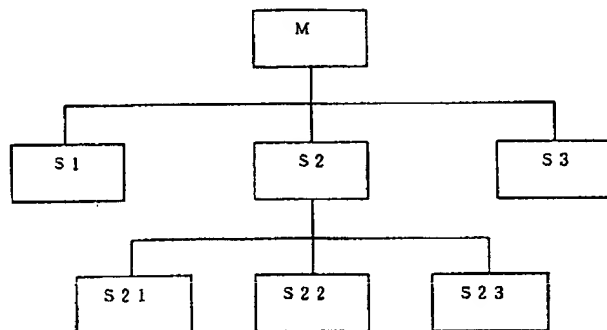
第 1 4 図



第 1 5 図



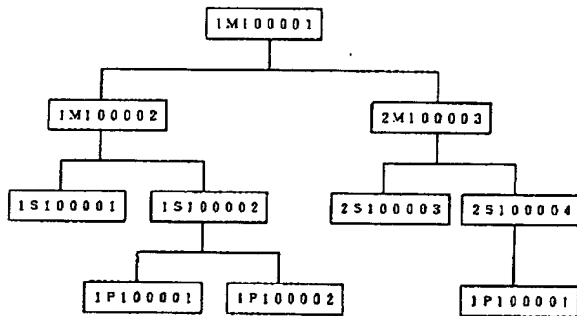
第 1 6 図



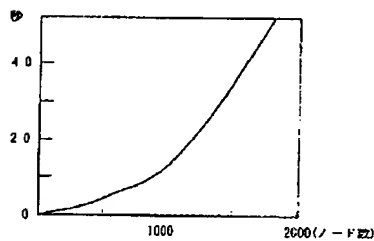
第 1 7 図

SEQ	LEVEL	部品番号	部品名称
1	1	1M100001	メインフレーム
2	2	1M100002	サブフレーム
3	3	1S100001	プレート1
4	3	1S100002	プレート2
5	4	1P100001	ボルト1
6	4	1P100002	ボルト2
7	2	1M100003	サブフレーム
8	3	2S100003	プレート3
9	3	2S100004	プレート4
10	4	1P100001	ボルト1

第 1 8 図



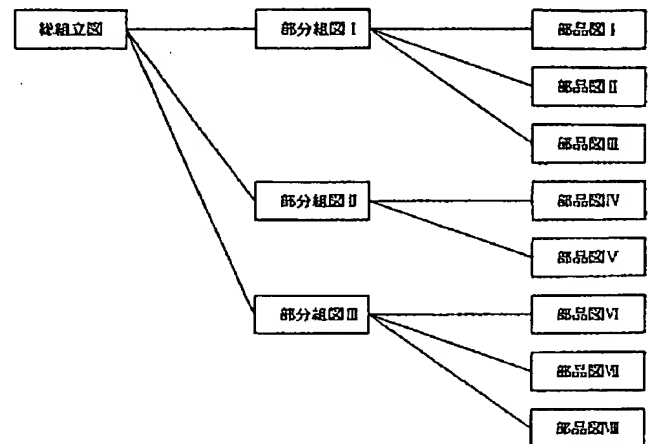
第 2 1 図



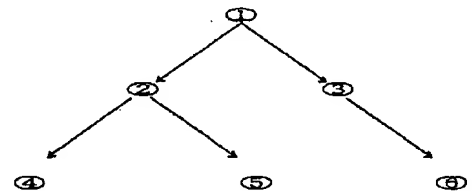
第 2 2 図

ノード ID	ノード名	親 ノード数	親ノードアドレス			子 ノード数	子ノードアドレス		
			ID	ID	ID		ID	ID	ID
01	1P10002	0				3	02	03	04
02	1P100021	1	01			0			
03	1P100022	1	01			0			
04	1P100023	1	01			0			

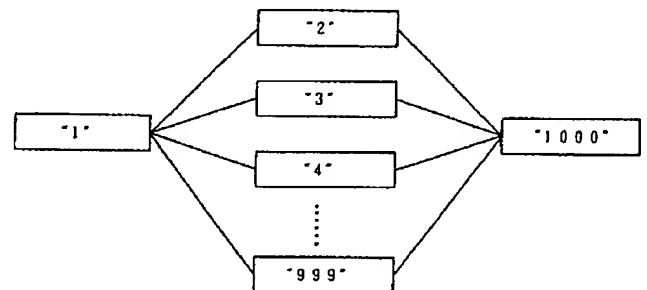
第 1 9 図



第 2 0 図



第 2 3 図



第 2 4 図

